

Bab 11. Aliran dan Pergerakan Masyarakat

## 2.1. Penelitian Terkait

Penelitian yang dilakukan oleh Zhang dan Zhao (2008) yang membandingkan pengendali *Proportional Integral* (PI), *Sliding Mode Control* (SMC), *Fuzzy Logic Controller* (FLC) dengan mengamati empat aspek perbandingan yaitu kinerja dinamis, akurasi *steady state*, ketahanan terhadap perubahan parameter dan implementasi. Dari keempat pengendali tersebut terbukti bahwa pengendali SMC dan FLC memberikan kinerja lebih yang bagus dari pada pengendali PI, tetapi untuk di implementasikan pengendali SMC jauh lebih mudah dibandingkan PI dan FLC. SMC mempunyai kemampuan yang lebih baik dalam menghadapi gangguan terhadap berbagai gangguan eksternal, perubahan parameter dan ketidakpastian sistem.

Dian (2013) dalam penelitiannya merancang pengendali *Proportional Integral* (PI) di dalam permukaan luncur *sliding mode* untuk mengendalikan level dan konsentrasi pada proses sistem CSTR. Penelitiannya menunjukkan bahwa pengendali *sliding mode* dengan permukaan luncur PI memiliki performansi yang lebih baik dari pada pengendali *sliding mode* saja. Hal ini dapat dilihat dari waktu *transient* yang cepat, kekokohan dalam menghadapi gangguan, dan meminimumkan *error steady state*. Pengendali PI digunakan untuk menghilangkan *error steady state* yang besar akibat terjadinya fenomena *chattering*.

Selanjutnya penelitian Dian (2014), dengan judul Pengendalian *Level* dan Konsentrasi pada *Continuous Stirred Tank Reactor* (CSTR) menggunakan *Sliding Mode Control* dengan permukaan luncur *Proportional Derivative* (PD). Dian merancang



pengendali PD di dalam permukaan luncur *sliding mode*, kemudian hasil perancangannya dibandingkan dengan SMC dan PID. Dari penelitian yang dilakukan dapat disimpulkan bahwa Pengendali *sliding mode* dengan permukaan luncur PD tidak berhasil membuat pengendalian *level* mencapai *setpoint* yang diinginkan. Pada saat diberikan gangguan, pengendali *sliding mode* dengan permukaan luncur PD menghasilkan kinerja yang kurang baik dibandingkan pengendali *sliding mode* dan PID, tetapi SMC dengan permukaan luncur PD dapat menghilangkan *overshoot* dibandingkan dengan pengendali SMC.

Penelitian Boy Putra (2014), dengan melakukan perancangan pengendali *Proportional Integral Derivative* (PID) didalam permukaan luncur *sliding mode* untuk mengendalikan *level* dan konsentrasi pada proses sistem CSTR. Penelitiannya menunjukkan bahwa pengendali yang dia rancang memiliki performansi yang paling baik jika dibandingkan dengan pengendali *sliding mode* dan *sliding mode* dengan permukaan luncur PI. Hal ini dibuktikan dari waktu *transient* yang cepat dalam mencapai *setpoint*, kemampuan menjaga kestabilan dan kekokohan dalam menghadapi gangguan dan *error steady state* yang kecil.

Penelitian Selanjutnya oleh Anton (2014) dengan melakukan analisa komparasi antara pengendali *static* dan *dynamic sliding mode*. Anton juga mengendalikan *level* dan konsentrasi pada proses sistem CSTR. Dari hasil penelitiannya, menunjukkan bahwa pengendali *dynamic sliding mode* memiliki performansi yang lebih baik daripada *static sliding mode*. Hal ini didukung oleh Slotine dan Li (1991) dan penelitian Teguh (2010), yang menunjukkan bahwa *dynamic sliding mode* memiliki kekokohan dalam menghadapi gangguan internal maupun eksternal yang lebih baik daripada *static sliding mode*.

Pemilihan pengendali *sliding mode* didasari karena pengendali ini mempunyai sifat yang kokoh dalam menghadapi gangguan yang telah dibuktikan oleh peneliti-peneliti sebelumnya. Namun pengendali *sliding mode* memiliki kekurangan yaitu terjadinya fenomena *chattering*. Hal ini terjadi karena pengendali *sliding mode* merupakan kendali umpan balik yang memiliki pensaklaran berkecepatan tinggi, sehingga menimbulkan osilasi yang berfrekuensi tinggi. Fenomena *chattering* ini dapat berdampak terhadap mekanis sistem, hilangnya energi pada *power*, menimbulkan *error steady state*, ketidakstabilan sistem dan menyebabkan akurasi menjadi lemah (Slotine dan Li (1991), Dian (2013)).

Fenomena *chattering* ini dapat dikurangi dengan fungsi *boundary layer* atau menggantikan fungsi *Signum* (Sgn) menjadi fungsi *Saturation* (Sat) (Slotine dan Li (1991),



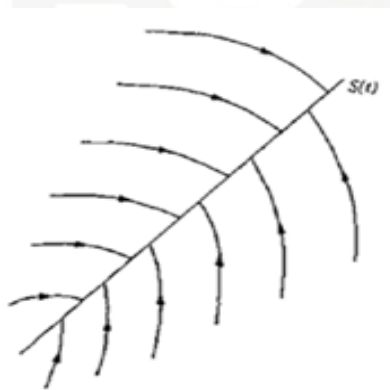
Kim, dkk (2006), Levant (2010)). Namun hal ini hanya dapat mengurangi fenomena *chattering* itu sendiri, sedangkan persoalan *error steady state* dan kestabilan sistem belum mendapat solusi yang tepat. Dari penelitian yang dilakukan oleh Boy Putra (2014) dan di dalam buku Ogata (1970), mengatakan bahwa penambahan pengendali PID dapat mengatasi persoalan *error steady state* dan ketidakstabilan sistem yang terjadi.

Pada penelitian ini, peneliti akan melakukan perancangan *hybrid* pengendali *dynamic sliding mode* dan pengendali *sliding mode* dengan permukaan luncur PID mengendalikan *level* dan konsentrasi pada sistem CSTR untuk mendapatkan performansi yang baik dalam mengatasi *error steady state*, ketidakstabilan sistem, dan gangguan yang diberikan dengan kemampuan yang lebih baik.

## 2.2. Dasar Teori

### 2.2.1. Pengendali *Sliding Mode*

Pengendali *sliding mode* merupakan sebuah kendali umpan balik pensaklaran berkecepatan tinggi (*high speed switching feedback control*) yang efektif dan kokoh dalam mengendalikan sistem linier maupun non linier (Slotine dan Li, 1991). Pengendali *sliding mode* terkenal karena kekokohnya karena pengendali ini menyediakan sebuah metode perancangan sistem yang tidak peka terhadap ketidakpastian parameter lingkungan dan gangguan dari luar (Slotine dan Li, 1991).



Gambar 2.1 Kondisi *Sliding*

Sumber : Slotine dan Li (1991)

Pengendali *sliding mode* ini telah banyak diterapkan dalam berbagai sistem seperti electronic power, power konverter, proses industri dan lain sebagainya. Prinsip kerja dari *Sliding Mode Control* ini yaitu pengendali ini memaksa trajektori state sistem menuju permukaan luncur (*Sliding Surface*) tertentu agar tetap meluncur pada permukaan tersebut,





tetapi keadaan ini mengakibatkan terjadinya osilasi pada permukaan luncur. Osilasi ini disebut fenomena *chattering*, *chattering* ini dapat menimbulkan *error steady state* (Slotine dan Li, 1991).

### 2.2.2. Permukaan Luncur

Permukaan luncur merupakan komponen penting pada dari pengendali *sliding mode* sebagai trayektori keadaan meluncur dari kondisi awal (*initial condition*) menuju keadaan yang diinginkan (*reference point*). Permukaan luncur (*sliding surface*) adalah suatu permukaan, lintasan atau trayektori yang dipilih dengan pertimbangan status trayektori sistem dapat menuju permukaan tersebut di manapun kondisi awalnya dalam waktu yang terbatas, serta status trayektori sistem dapat dipertahankan di sekitar permukaan luncur tersebut (Slotine dan Li, 1991).

Permukaan luncur didefinisikan sebagai  $\sigma(x)$  atau  $S$  :

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e \quad (2.1)$$

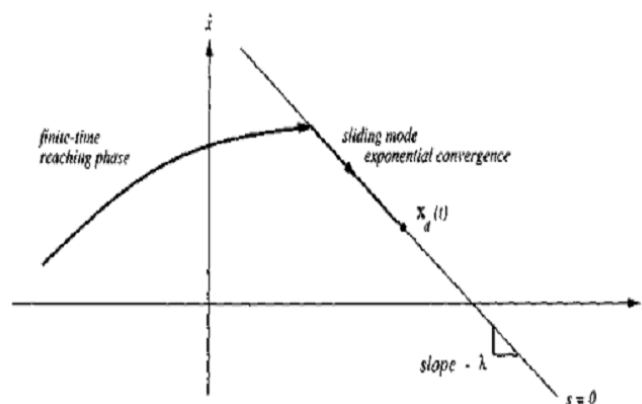
Dimana:

$n$  : orde sistem

$e$  : error

$\lambda$  : konstanta positif

Ilustrasi permukaan luncur pada pengendali *sliding mode* ditunjukkan pada gambar 2.2.



Gambar 2.2 Permukaan Luncur pada Sliding Mode

Sumber : Slotine dan Li (1991)

Pengendali *sliding mode* dapat dirancang dengan memodifikasi permukaan luncur sesuai dengan kebutuhan sistem dan hasil yang ingin dicapai.



### 2.2.3. Perancangan Permukaan Luncur

Slotine dan Li (1991) dalam penelitiannya menuliskan bahwa perancangan permukaan luncur dapat dirumuskan dengan persamaan awal sebagai berikut :

Jika terdapat suatu sistem dinamis :

$$\dot{x} = f(x, t) + b(x, t)u + d(t) \quad (2.2)$$

dimana  $u$  adalah *control input*,  $x$  merupakan vektor keadaan,  $f(x, t)$  dan  $b(x, t)$  berupa fungsi terbatas,  $d(t)$  adalah gangguan eksternal. Jika  $x_{ref}$  merupakan nilai yang diinginkan (*set point*) maka *error* dapat dinyatakan dengan :

$$e(t) = x(t) - x_{ref}(t) \quad (2.3)$$

fungsi *switching* yaitu permukaan luncur  $S(x, t)$  untuk *static sliding mode* :

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e \quad (2.4)$$

permukaan luncur untuk *dynamic sliding mode* :

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n+1-1} e \quad (2.5)$$

Permukaan luncur untuk PID *sliding mode* :

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e + \lambda_1 \int_0^t e(t) dt + \lambda_2 \frac{de}{dt} \quad (2.6)$$

Dengan  $\lambda$  merupakan suatu konstanta positif. Dimana fungsi *switching* ini digunakan untuk menentukan besarnya nilai  $u$  agar memenuhi kondisi *sliding*. Permukaan *sliding* ini membagi bidang menjadi dua bagian, yaitu  $S > 0$  dan  $S < 0$ . Agar *trajectory* keadaan dapat meluncur, maka sistem harus berada dalam kondisi *sliding*. Dengan demikian, besar nilai *input* kendali bergantung pada nilai  $S(x, t)$ . Perancangan *control law* (aturan kendali) pada SMC dilakukan hingga  $e(t)$  bergerak menuju permukaan *sliding* dan meluncur pada permukaan tersebut untuk semua  $t \geq 0$ .

Permukaan luncur merupakan persamaan yang memenuhi :

$$S(x, t) = 0 \quad (2.7)$$

besarnya nilai *control input* pada pengendali *sliding mode* bergantung pada nilai  $S$ , sehingga disebut kondisi *sliding*. Kondisi tersebut harus memenuhi persamaan lyapunov.

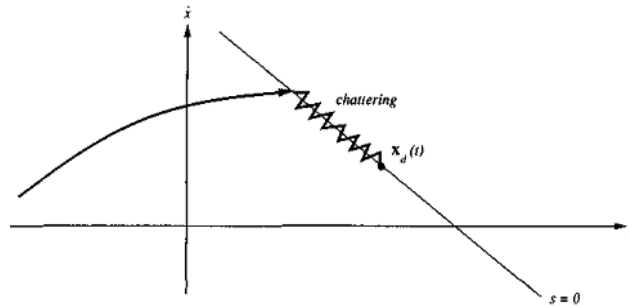
$$\dot{S} \leq -\eta |S| \quad (2.8)$$

Dengan  $\eta$  suatu konstanta positif.



#### 2.2.4. Chattering

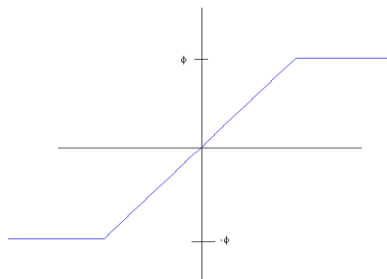
Pengendali *sliding mode* termasuk dalam pengendali berumpan balik dengan pensaklaran berkecepatan tinggi (*high speed switching feedback*), sehingga dalam penerapannya *sliding mode pengendali* memiliki kelemahan yaitu kondisi *chattering* yang dapat mengganggu kestabilan sistem. *Chattering* merupakan osilasi keluaran pengendali dengan frekuensi tinggi yang disebabkan oleh *switching* yang sangat cepat untuk membentuk *sliding mode*. Osilasi yang sangat tinggi pada sinyal kendali ini menyebabkan ketidakstabilan pada sistem. Fenomena *chattering* dapat dilihat pada gambar 2.3.



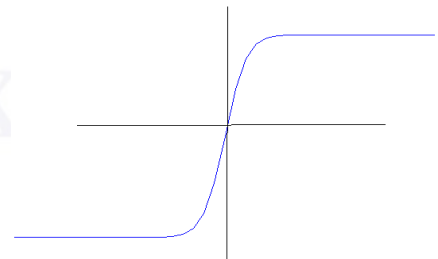
Gambar 2.3 Fenomena *chattering*

Sumber : Slotine dan Li (1991)

*Chattering* yang besar akan mengakibatkan akurasi menjadi rendah, rusaknya sistem mekanis aktuator, hilangnya energi pada *power, error steady state*, ketidakstabilan sistem dan lain sebagainya. Berbagai cara dapat dilakukan untuk mereduksi *chattering*, namun yang paling efektif adalah mengubah fungsi diskontinyu *signum* menjadi fungsi kontinyu saturasi / histeresis / *arcus tangen* (Slotine dan Li (1991), Kim, dkk (2006), Levant (2010)).



a. Fungsi Saturasi



b. Fungsi Arcus Tangen

Gambar 2.4 Fungsi saturasi dan Arcus Tangen

Sumber : Dian (2013)



### 2.2.5. Sinyal Kendali

Untuk mendapatkan sinyal kendali yang mampu membawa status *variabel* menuju permukaan luncur dan mempertahankan status *trayektori* agar tetap berada di sekitar permukaan luncur, maka diperlukan dua macam sinyal kendali. Sinyal kendali yang pertama adalah sinyal kendali ekivalen yang berfungsi untuk membawa status *variabel* menuju permukaan luncur ( $U_{eq}$ ) dan sinyal kendali yang kedua adalah sinyal kendali natural yang berfungsi untuk mempertahankan status *variabel* agar tetap berada di sekitar permukaan luncur ( $U_n$ ) (Gwo Ruey, dkk., 2004).

Untuk mendapatkan sinyal kendali *sliding mode* dilakukan beberapa langkah sebagai berikut :

1. Misalkan terdapat sebuah sistem linier  $\dot{x} = Ax + Bu$
2. untuk mendapatkan  $u_{eq}$  disain permukaan luncur dengan  $S(x,t) = 0$

$$SAx + SBu_{eq} = 0$$

$$SBu_{eq} = -SAx$$

$$u_{eq} = -(SB)^{-1} SAx$$

3. Setelah  $u_{eq}$  diperoleh, berdasarkan persamaan lyapunov dapatkan  $u_n$  :

$$\dot{S} = SAx + SBu_{eq} + SBu_n$$

$$\dot{S} = SAx + S(-(SB)^{-1} SAx) + SBu_n$$

$$\dot{S} = SAx + SB(SB)^{-1} SAx + SBu_n$$

$$\dot{S} = SAx - SAx + SBu_n$$

$$\dot{S} = SBu_n$$

4. Untuk memenuhi kondisi *sliding*, maka :

$$SBu_n = -Ksign(S)$$

$$U_n = -(SB)^{-1} Ksign(S)$$

5. Diperoleh sinyal kendali total sebagai berikut :

$$u = u_{eq} + u_n$$

$$u = -(SB)^{-1} SAx - (SB)^{-1} Wsat(\sigma(x))$$

Sinyal kendali total merupakan penjumlahan dari dua sinyal kendali tersebut dan dapat ditulis:

$$\bar{U} = U_{eq} + U_n \quad (2.9)$$





### 2.2.6. Kestabilan Lyapunov

Alexander Mikhailovitch Lyapunov, seorang ilmuwan dari Rusia, telah mengembangkan metode untuk menentukan stabilitas dari sistem proses didasarkan dari penghematan energi yang disimpan. Sistem dikatakan stabil apabila energi yang disimpan makin lama makin kecil, maka osilasi yang terjadi juga semakin lama semakin kecil. Sebaliknya, sistem dikatakan tidak stabil apabila energi yang disimpan makin lama makin besar, maka osilasi yang terjadi semakin lama semakin besar juga. Agar sistem dapat dianalisis kestabilannya, maka perlu dibuat model matematis yang menghubungkan antara masukan, proses dan keluaran (Slotine dan Li, 1991).

Pada sistem pengendalian, diharapkan agar keluaran/respon dari sistem akan menuju nilai yang sama dengan masukan/*set point*. Hal ini identik dengan mengharapkan agar nilai *error* dapat menuju nilai nol serta tetap berada pada nilai nol. Mengacu pada teori *state-space*, di mana nilai *error* dapat diambil sebagai variabel *state* sistem, maka dapat juga diambil suatu fungsi skalar :

$$V(x) = \frac{1}{2} [S]^2 \quad (2.10)$$

yang memenuhi :

$$\dot{V}(x) = S\dot{S} < 0 \quad (2.11)$$

### 2.2.7. Continuous Stirred Tank Reactor (CSTR)

CSTR adalah tangki pengaduk yang umumnya digunakan pada industri-industri kimia yang memerlukan pencampuran dua fluida atau lebih. CSTR merupakan sistem yang non linier multivariabel, dikatakan multivariabel karena banyaknya variabel pada CSTR yang dapat dikendalikan antara lain tekanan (*pressure*), suhu (*temperature*), laju aliran (*flow*), ketinggian (*level*), konsentrasi (*concentration*), dan lain-lain. CSTR dapat ditemukan pada industri makanan, minuman, bioteknologi, dan sebagainya.



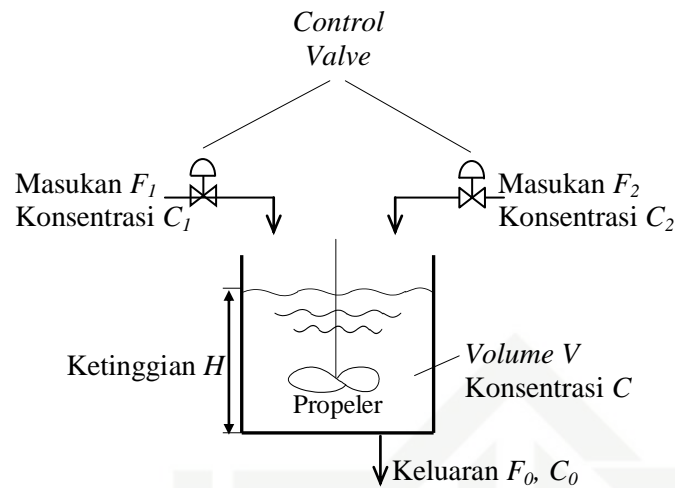
# Hak Cipta Dilindungi Undang-Undang

1. Diarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Diarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.5 Dinamika Sistem CSTR

Sumber : Dian (2013)

Berdasarkan hukum kesetaraan *volume* tangki dengan asumsi tidak adanya material yang keluar dalam bentuk uap, maka pemodelan matematis sistem dapat ditulis sebagai berikut :

$$V_{in} - V_{out} = V_{dalam\ tangki} \quad (2.12)$$

$$(F_1\Delta t + F_2\Delta t) - F_0\Delta t = \Delta V \quad (2.13)$$

$$\frac{\Delta V}{\Delta t} = F_1 + F_2 - F_0 \quad (2.14)$$

*Volume* dalam tangki merupakan hasil perkalian dari luas permukaan tangki (A) dengan perubahan *level* (H) dalam tangki, sehingga dapat ditulis:

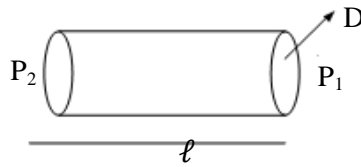
$$\Delta V = A\Delta H \quad (2.15)$$

Lalu substitusikan persamaan 2.15 ke persamaan 2.14, sehingga persamaan menjadi seperti di bawah ini:

$$\frac{A\Delta H}{\Delta t} = F_1 + F_2 - F_0 \quad (2.16)$$

$$\frac{dH}{dt} = \frac{1}{A}(F_1 + F_2) - \frac{1}{A}F_0 \quad (2.17)$$

Kecepatan aliran keluaran  $F_0$  tergantung dari ketinggian (H) permukaan dalam tangki, dan luas diameter pipa (Dian, 2013). Nilai konstanta celah didapat dari ilustrasi berikut ini :



Gambar 2.6 Pipa Keluaran CSTR

Sumber : Dian (2013)

Berdasarkan ilustrasi pada gambar 2.6 dapat dituliskan :

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{P_1 - P_2} \quad (2.18)$$

Dimana:  $P_1 \gg P_2$

$$P = \rho g H$$

$$F_0 = \text{laju aliran } (m^3/s)$$

$$D = \text{diameter pipa } (m)$$

$$\ell = \text{panjang pipa } (m)$$

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{P_1} \quad (2.19)$$

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{\rho g H} \quad (2.20)$$

$$F_0 = \frac{Dk_t \sqrt{\rho g H}}{P\ell} \quad (2.21)$$

dengan demikian, kecepatan aliran keluaran  $F_0$  tergantung dari ketinggian permukaan dalam tangki ( $H$ ), luas diameter pipa ( $D$ ), dan konstanta celah, sehingga :

$$\frac{Dk_t \sqrt{\rho g}}{P\ell} = Kc \quad (2.22)$$

Sehingga persamaan 2.21 menjadi seperti di bawah ini:

$$F_0 = Kc \sqrt{H} \quad (2.23)$$

Substitusikan persamaan 2.23 ke persamaan 2.17, sehingga didapat pemodelan matematis untuk pengendalian *level* sebagai berikut :

$$\frac{dH}{dt} = \frac{1}{A} (F_1 + F_2) - \frac{Kc}{A} \sqrt{H} \quad (2.24)$$

dimana:

$$A = \text{Luas permukaan tangki } (m^2)$$

$$F_1 = \text{Laju aliran 1 } (m^3/s)$$



$F_2$  = Laju aliran 2 ( $m^3/s$ )

$K_c$  = Konstanta celah

$H$  = Ketinggian fluida ( $m$ )

Berdasarkan kesetaraan massa dari ilustrasi gambar 2.5 didapat pemodelan matematis untuk konsentrasi di dalam tangki adalah sebagai berikut:

$$\Delta C_0 V = C_1 F_1 \Delta t + C_2 F_2 \Delta t - C_0 F_0 \Delta t \quad (2.25)$$

$$\frac{\Delta C_0 V}{\Delta t} = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.26)$$

$$\frac{dC_0 V}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.27)$$

Dimana,  $\frac{dUV}{dt} = U'V + UV'$  maka;

$$\frac{dC_0 V}{dt} = V \frac{dC_0}{dt} + C_0 \frac{dV}{dt} \quad (2.28)$$

Substitusikan persamaan 2.27 ke persamaan 2.28.

$$V \frac{dC_0}{dt} + C_0 \frac{dV}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.29)$$

$$V \frac{dC_0}{dt} + C_0 (F_1 + F_2 - F_0) = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.30)$$

$$V \frac{dC_0}{dt} + C_0 F_1 + C_0 F_2 - C_0 F_0 = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.31)$$

$$V \frac{dC_0}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 - C_0 F_1 - C_0 F_2 + C_0 F_0 \quad (2.32)$$

$$V \frac{dC_0}{dt} = [(C_1 - C_0)F_1 + (C_2 - C_0)F_2] \quad (2.33)$$

Karena  $V = AH$  maka;

$$AH \frac{dC_0}{dt} = [(C_1 - C_0)F_1 + (C_2 - C_0)F_2] \quad (2.34)$$

$$\frac{dC_0}{dt} = \frac{1}{AH} [(C_1 - C_0)F_1 + (C_2 - C_0)F_2] \quad (2.35)$$

Sehingga didapat persamaan untuk pengendalian konsentrasi pada CSTR dengan konstanta pengaduk ( $K_p$ ) adalah sebagai berikut:

$$\frac{dC_0}{dt} = \frac{K_p}{AH} [(C_1 - C_0)F_1 + (C_2 - C_0)F_2] \quad (2.36)$$



Dimana:

|       |   |       |   |
|-------|---|-------|---|
| $H$   | = ketinggian fluida (m)                 | $F_1$ | = laju aliran fluida 1 ( $m^3/detik$ )    |
| $C_0$ | = konsentrasi fluida ( $kmol^3/detik$ ) | $F_2$ | = laju aliran fluida 2 ( $m^3/detik$ )    |
| $A$   | = luas tangki ( $m^2$ )                 | $C_1$ | = konsentrasi fluida 1 ( $kmol^3/detik$ ) |
| $K_p$ | = konstanta pengaduk                    | $C_2$ | = konsentrasi fluida 2 ( $kmol^3/detik$ ) |

Dari persamaan 2.24 dan 2.36 dapat dilihat bahwa dinamika fungsi CSTR adalah non linier. Data parameter proses CSTR dapat dilihat pada tabel 2.1.

Tabel 2.1. Parameter proses CSTR

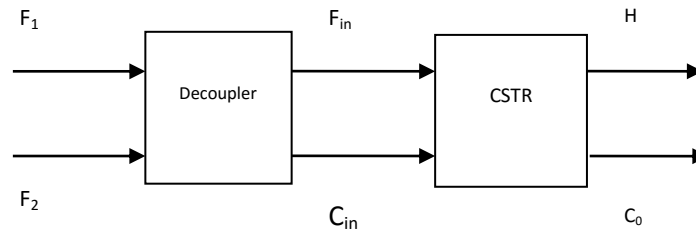
| Parameter                  | Simbol | Nilai     | Satuan     |
|----------------------------|--------|-----------|------------|
| Laju aliran 1              | $F_1$  | 0.15      | $m^3/s$    |
| Laju aliran 2              | $F_2$  | 0.6       | $m^3/s$    |
| Konsentrasi 1 (konstan)    | $C_1$  | 1         | $kmol^3/s$ |
| Konsentrasi 2 (bervariasi) | $C_2$  | 1.2 – 1.4 | $kmol^3/s$ |
| Volume                     | $V$    | 1         | $m^3$      |
| Luas permukaan tangki      | $A$    | 1         | $m^2$      |
| Konstanta celah (beban)    | $K_c$  | 0.5 – 1   | -          |
| Konstanta pengaduk         | $K_p$  | 0.2       | -          |

Sumber: Dian (2013)

### 2.2.8. Decoupler

Dari penelitian sebelumnya, pada sistem CSTR antara *level* dan konsentrasi itu saling mempengaruhi, untuk itu dilakukan *decouple* agar *level* dan konsentrasi dapat dikendalikan secara terpisah. *Decouple* dilakukan dengan merancang suatu sistem baru yang membuat seolah-olah masukan sistem adalah  $F_{in}$  dan  $C_{in}$ . Desain sistem baru ini selanjutnya disebut dengan *decoupler*. Ilustrasi sistem dengan *decoupler* dapat dilihat pada gambar berikut (Dian, 2013):





Gambar 2.7 Desain Sistem Baru (*Decoupler*)  
(Sumber : Dian, 2013)

Untuk merancang sistem baru (*decoupler*), definisikan sistem sebagai berikut :

$$F_{in} = F_1 + F_2 \quad (2.37)$$

$$C_{in} = \frac{C_1 F_1 + C_2 F_2}{F_1 + F_2} \quad (2.38)$$

Keterangan :

$F_{in}$  = Sinyal kendali untuk pengendalian *level*

$C_{in}$  = Sinyal kendali untuk pengendalian konsentrasi

dimana  $F_1$  ;  $F_2$  sebagai fungsi  $C_{in}$  dan  $F_{in}$ .

untuk konsentrasi  $C_1$  :

$$C_1 F_1 + C_1 F_2 = C_1 F_{in} \quad (2.39)$$

$$C_1 F_1 + C_2 F_2 = C_{in} F_{in} -$$

$$F_2 = \frac{1}{(C_1 - C_2)} \{ C_1 F_{in} - C_{in} F_{in} \}$$

Untuk konsentrasi  $C_2$  :

$$C_2 F_1 + C_2 F_2 = C_2 F_{in}$$

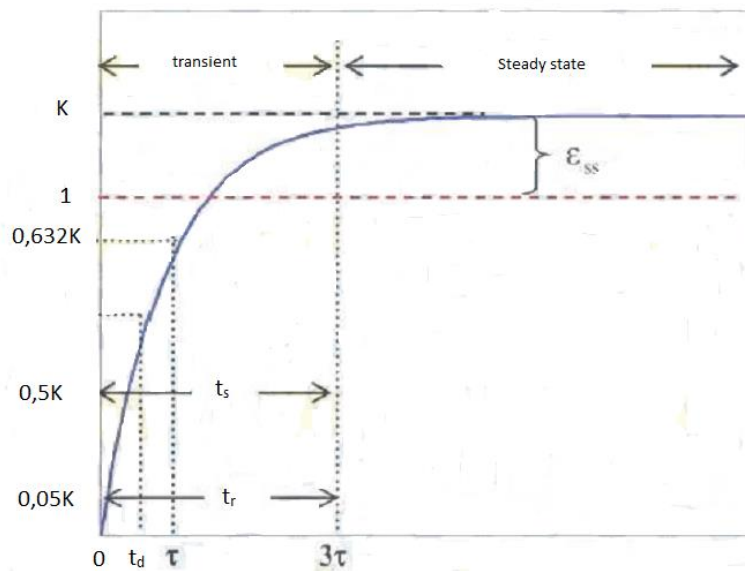
$$C_1 F_1 + C_2 F_2 = C_{in} F_{in} -$$

$$F_1 = \frac{1}{(C_2 - C_1)} \{ C_2 F_{in} - C_{in} F_{in} \} \quad (2.40)$$

## 2.2.9. Identifikasi Sistem

CSTR merupakan sistem berorde satu, oleh sebab itu metode identifikasi yang dilakukan adalah metode identifikasi statis. Metode identifikasi dilakukan dengan pendekatan grafis, di mana sinyal uji diberikan pada sistem untuk mengetahui respon *open loop* sistem. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari sistem.

Salah satu metode identifikasi statis adalah metode pengamatan respon waktu. Identifikasi sistem dengan metode ini bekerja berdasarkan pengamatan grafis terhadap masukan *step*. Karakteristik respon waktu untuk sistem orde pertama diberikan berdasarkan respon sistem terhadap masukan sinyal *step*. Karakteristik respon waktu sistem orde pertama dibedakan menjadi karakteristik respon transien dan keadaan tunak (*steady state*). Grafik respon sistem orde pertama untuk  $X_{ss} = 1$  dan  $Y_{ss} = K$  ditunjukkan pada Gambar 2.7.



Gambar 2.8. Respon Sistem Orde Satu

Sumber: Ogata (1970)

Respon transien terdiri dari (Ogata, 1970):

a) Spesifikasi teoritis :

Konstanta waktu ( $\tau$ ), adalah waktu yang dibutuhkan respon mulai dari  $t=0$  sampai dengan respon mencapai 63,2% dari respon *steady state*. Konstanta waktu menyatakan kecepatan respon sistem. Konstanta waktu yang lebih kecil akan mempercepat respon sistem

b) Spesifikasi praktis :

1. Waktu tunak atau *settling time* ( $t_s$ ), adalah ukuran waktu yang menyatakan bahwa respon sistem telah masuk pada daerah stabil. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_s$  dapat diformulasikan sebagai berikut :

$$t_s (\pm 3\%) \approx 5\tau$$

$$t_s (\pm 4\%) \approx 2\tau$$



#### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber.

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$t_s (\pm 5\%) \approx 0,5\tau$$

- Waktu naik atau *rise time* ( $t_r$ ), adalah ukuran waktu yang menyatakan bahwa respon sistem telah naik dari 5% ke 95% atau 10% ke 90% dari nilai respon pada keadaan tunak (*steady state*). Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_r$  dapat diformulasikan sebagai berikut :

$$t_r (5\% - 95\%) \approx \tau \ln 19$$

$$t_r (10\% - 90\%) \approx \tau \ln 9$$

- Waktu tunda atau *delay time* ( $t_d$ ), adalah waktu yang dibutuhkan respon mulai  $t=0$  sampai respon mencapai 50% dari nilainya pada keadaan tunak (*steady state*). Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_d$  dapat diformulasikan sebagai berikut :

$$t_d (\pm 3\%) \approx \tau \ln 2$$

- Karakteristik respon keadaan tunak (*steady state*) sistem orde pertama diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* ( $e_{ss}$ ).

$$e_{ss} = R_{ss} - C_{ss}$$

dengan  $C_{ss}$  dan  $R_{ss}$  masing – masing adalah keluaran dan masukan sistem pada keadaan tunak.

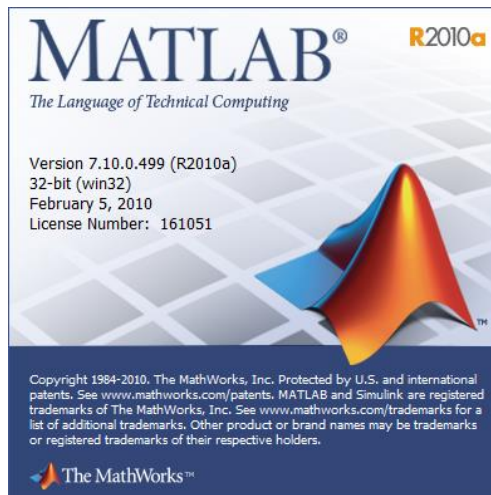
- Nilai *overshoot* yang diperbolehkan adalah 2 % dari *setpoint*.

### 2.2.10. Perangkat lunak MATLAB

Matlab merupakan singkatan dari *Matrix Laboratory* yang berarti bahasa pemrograman tingkat tinggi (semakin tinggi tingkat bahasa pemrograman maka semakin mudah cara penggunaannya) dengan kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan tunggal. Matlab memberikan sistem interaktif yang menggunakan konsep *array/matrix* sebagai variabel elemennya tanpa membutuhkan pendeklarasian *array*.

#### Hak Cipta Dilindungi Undang-Undang

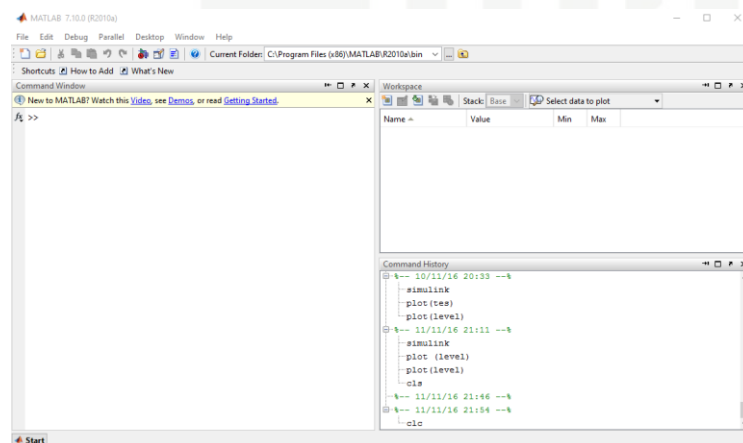
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.9. Tampilan awal MATLAB R2010a

Matlab dikembangkan oleh Mathwork pada tahun 1970. Aplikasi Matlab ini banyak digunakan dalam bidang yang membutuhkan perhitungan matematika yang rumit, dimana seluruh operasi perhitungan dalam Matlab berupa operasi matrik. Matlab dapat menampilkan hasil perhitungan dalam bentuk *plot* grafik. Pada perangkat lunak Matlab terdapat beberapa bagian penting yang digunakan dalam menjalankan program, yaitu:

1. *Command window* digunakan untuk mengetik fungsi yang diinginkan.
2. *Command history* berfungsi agar fungsi yang telah digunakan sebelumnya dapat digunakan kembali.
3. *Workspace* digunakan untuk membuat variabel yang ada dalam Matlab.



Gambar 2.10. Tampilan Matlab R2010a

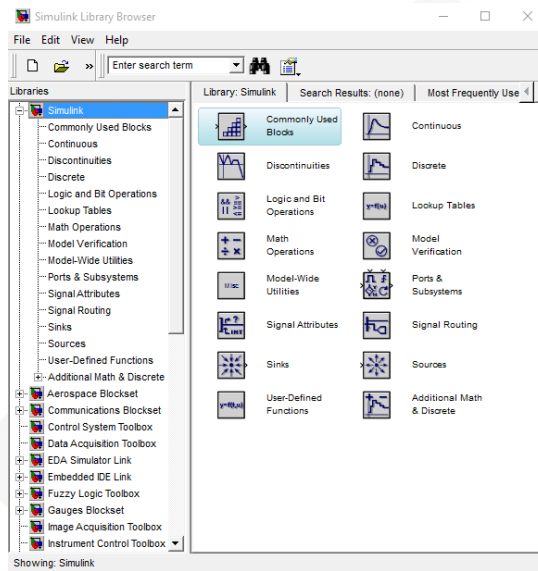
*Simulink* adalah sebuah kumpulan aplikasi dalam Matlab untuk melakukan modeling, simulasi, dan untuk melakukan analisis dinamik pada suatu sistem. Program *simulink*





memudahkan *user* untuk membuat suatu simulasi lebih interaktif. Tiruan sistem diharapkan mempunyai perilaku yang sangat mirip dengan sistem fisik. Jika digunakan dengan benar, simulasi akan membantu proses analisi dan desain sistem.

*Simulink* dalam Matlab juga dapat menunjukkan performansi sistem dalam bentuk dua ataupun tiga dimensi. Dalam perancangannya *user* menjadi mudah karena adanya blok-blok diagram yang dapat dengan mudah diatur sedemikian rupa, sesuai dengan model matematis dari sistem atau *plant* yang akan dikendalikan.



Gambar 2.11. Kotak dialog *Simulink Library Browser*